## Approximating Safe Spacing Policies for Adaptive Cruise Control Strategies

van Willigen, W.H.; Schut, M.C.; Kester, L.J.H.M.

***published in***
2011 IEEE Vehicular Networking Conference (VNC)
2011

# Approximating Safe Spacing Policies for Adaptive Cruise Control Strategies

W.H. van Willigen
VU University Amsterdam (NL)
and TNO The Hague (NL)
w.h.vanwilligen@vu.nl

M.C. Schut
VU University Amsterdam (NL)
m.c.schut@vu.nl

L.J.H.M. Kester
TNO
The Hague (NL)
leon.kester@tno.nl

***Abstract*—In the development of Cooperative Adaptive Cruise Control (CACC) systems, spacing policies are primarily developed for optimisation of string stability and traffic stability. However, the safety issue is hardly taken into account. Uncertainty in the communication network and sensor information makes deciding upon a safe minimal headway a non-trivial task. In this paper, we propose a model that is able to approximate the minimal safe time headway, given uncertainty of parameters with varying velocities. By simulating emergency stops, we use the difference in displacement of the cars and a desired maximum probability of a crash to approximate the minimum time headway that yields this probability of a crash. The resulting method is necessary for platooning, a major research development in vehicular networking systems.**

## I. Introduction

Cooperative Adaptive Cruise Control (CACC) is one of the major developments in recent research on Vehicular Networking (VN). Its predecessor, i.e. ACC, is currently successfully being deployed commercially by automotive manufacturers, and CACC is expected to shortly ship to the market as well. The proposed *cooperation* in CACC means that vehicles have a wireless vehicle-to-vehicle communication system (which then needs a new control logic) in addition to an existing ACC system.

The main challenge for CACC is to drive comfortably and safely at time headways that are significantly smaller than what is currently possible with human drivers (which is, typically, one second). The time headway is the distance measured in time between vehicles in a transit system. It is one of the pivotal parts of the so-called *spacing policy* in a cruise control system, which refers to the desired steady state distance that an (C)ACC system attempts to maintain from the preceding vehicle.

With the addition of a communication system, there are also many *uncertainties* introduced in the control system, in addition to the already existing uncertainties from the radar. Such uncertainties can come from, among others, packet loss distributions, vehicle modelling errors, and driver behaviour models. These uncertainties need to be incorporated when determining appropriate spacing policies. But there are also specific *dynamics* that we need to consider because of the variety in surrounding vehicles and changing environmental conditions (e.g. road network, weather). The combination of these uncertainties and dynamics make it impossible to completely determine good spacing policies (and time headways) beforehand. We thus need to move to variable (or: dynamic) policies instead of constant ones. We have seen a similar shift of research attention for ACC systems [1] 10-15 years ago, which lead to development of variable-time gap (time headway) policies instead of constant-time gap ones.

Much work in CACC focuses on the comfort issue (e.g. to create string-stable platoons), but we concentrate in this paper on the *safety* aspect. We aim to answer the question what are safe time headways *given current circumstances*. These circumstances then refer to the before-mentioned uncertainties and dynamics. In CACC, the coordinated safety management tries to guarantee safety in worst-case future developments (e.g. emergency braking). In this paper, we investigate these worst-case scenarios by means of computer simulations based on given (C)ACC models. The considered scenario includes two vehicles that are driving behind each other, where the first vehicle makes an emergency stop – we then look at the reaction of the second vehicle. On a side note, these simulations could be seen as a Monte Carlo algorithm that could later be used on-line (i.e. while the vehicle is driving) to dynamically determine spacing policies, but this is currently outside the scope of this paper.

While we thus consider determining time headways on-line as future work, the above arguments should be sufficiently compelling to completely abandon attempting to determine constant times beforehand (i.e. off-line) for CACC systems and we should move to try designing policies with *variable* time headways. In the experiments in this paper, we systematically investigate the effects of such variable times in different systems (ACC and two different CACC systems). This is the necessary ground work that needs to done before we can move to developing *on-line* methods determining variable-time spacing policies.

In our experimental approach, we assume certain parameters of uncertainty. Using these parameters, we experimentally derive the minimal safe time headway. Our results are therefore only valid for this particular set of uncertainty parameters. However, the main point we make in this paper, is that our *approach* is valid. In other words, our method will derive the minimal safe time headway for a particular set of uncertain parameters, and we illustrate this in this paper by experimenting with a particular instance of these parameters.

The remainder of this paper is structured as follows. In Section II, we discuss related literature. In Section III, we introduce the model that we used for our research. Section IV describes the experiments and results. We analyse these results in Section V, and we conclude the paper in Section VI.

## II. BACKGROUND

Our work is positioned with (C)ACC systems that address spacing policies based on variable and constant time headways, where performance is measured in terms of safety, comfort and traffic flow improvement. We focus in particular on modeling parametric uncertainties for CACC systems. We briefly overview relevant literature on these topics here.

Petrov [2] builds a non-linear adaptive tracking controller for a two-vehicle convoy, where the vehicles communicate neither with each other nor with the road infrastructure. Instead, standard robotic methodology is applied to do autonomous vehicle following, combined with a feedback-based controller (employed by the follower vehicle). This work assumes (actually, aims at) a prescribed inter-vehicle distance (what we call time headway).

An extensive review of constant time headways for ACC is done by Swaroop [1]. Three different performance criteria are considered: stability, safety, and traffic flow behaviour. For ACC, safety guarantees can be given, even such that errors in spacing do not amplify. Concerning stability and flow, smaller time headways are required to achieve higher throughputs. The review also shows that the control effort of an ACC system with a constant time headway is inversely proportional time headway: the smaller this time, the greater the control effort.

Parametric uncertainties in ACC systems have been researched by Swaroop [3]. These uncertainties concern vehicle mass, aerodynamic drag and time drag. The provided solution to address these uncertainties is a Lyapunov-based decentralised adaptive control algorithm.

In [4], Santhanakrishnan have developed a framework for design and evaluation of spacing policies for ACC. Although the evaluation criteria include string and traffic flow stability, and traffic flow capacity, the framework does not explicitly address safety.

Safety in ACC is an issue that is addressed explicitly by Wang and Rajamani [5]. In this work, an ACC system is proposed that can improve traffic flow and ensure safe operation. The novelty of the system is that it uses a new inter-vehicle spacing policy, in which the spacing is a non-linear function of vehicle speed (called the variable time-gap, VTG, policy). In comparison with a (then) traditional constant time-gap, CTG, policy, the same level of safety is provided, while improving the traffic flow. The question if ACC systems should in general be designed to maintain a constant time-gap between vehicles, is addressed in [6]. Another approach that improves CTG based system is described in [7]. Zhao et al. demonstrate a new spacing policy that is safe and improves traffic flow. The policy is a non-linear function of vehicle velocity and uses the vehicle state and braking capacity information. The policy works best in high-density traffic conditions.

Yi and Horwitz [8] propose an approach to macroscopic traffic flow propagation stability for ACC vehicles. In this approach, a non-linear traffic flow stability criterion is used with a wavefront expansion technique. In earlier approaches, a macro- with microscopic model was necessary with a constant time headway. The new approach covered all stability conditions obtained for these earlier approaches. Another VTG-policy based ACC system is proposed by Zhang [9]. This control system guarantees stability, and it regulates speed and separation errors toward zero (with the leading vehicle drives a constant speed).

While ACC systems are currently being adopted in consumer vehicles, research and development into cruise control focuses on enabling more and better cooperation between ACC systems, yielding so-called CACC systems. Van Arem *et al.* [10] describe the effect of CACC on traffic flow. They conclude that, when the penetration level of CACC-equipped vehicles is high enough ($> 60\%$), traffic stability and throughput is improved. In Yang *et al.* [11], a communication protocol is proposed in order to make a cooperative collision warning system on highways.

The main application area of CACC technology these days is platooning. Broggi *et al.* [12] and Kanellakopoulos [13] both use image recognition techniques in combination with sensors to autonomously enable platooning. However, current technology has improved significantly since then, and nowadays direct radio communication between vehicles is used to enable platooning.

Naus *et al.* [14] thoroughly investigate the issue of string stability in platooning, with both ACC and CACC controllers. Their method includes several factors of delay in communication, but uncertainty of the information is not taken into account.

In [15], an extensive architecture is given for a layered multi-agent CACC architecture. The authors use this architecture to implement both centralised platoons (in which there is a coordinating platoon leader) and decentralised platoons (in which all cars operate as equals). Khan *et al.* [16] present different platoon (in their paper, convoy) forming strategies, based on a utility value of a platoon.

To summarise, in all of the above approaches to designing CACC systems, uncertainty in information and communication is not accounted for. Also, the works focus on comfort (string stability) rather than safety. While these points have been addressed for ACC (as shown above in the first part of this section), this has not been picked up in CACC development and research. These are the points that we address in this paper: safety and parametric uncertainty in CACC systems. We build further on earlier work [17] where minimal safe time headways were experimentally determined for a number of different (C)ACC controllers.

## III. Model

In this section, we describe the model that we used for our simulations. This model is an extended version of the model that we introduced in [17]. It is a numerical model in which different kinds of uncertainty are explicitly modelled. This presence of uncertainty in information and communication justifies our choice for simulation, since it makes a mathematical analysis of the problem too complex.

First we describe how we modelled the cars in our simulation, and second, we describe the uncertainty in information and communication that we included in our model.

### A. Cars

We do experiments with three different types of car controller. First, there is the adaptive cruise control (ACC) controller. This controller uses the radar sensor of the car to derive information about distance, velocity and acceleration from the preceding vehicle. Second and third are the two cooperative adaptive cruise control controllers (CACC1 and CACC2). These controllers make use of direct communication between vehicles to derive the same information about preceding vehicles.

The difference in technology between the ACC controller and the CACC controllers has some implications. First, a radar sensor can only measure the range (direct distance to the preceding vehicle) and range rate (relative velocity of the preceding vehicle). This means that information about the acceleration of the preceding vehicle needs to be derived from this information. When using direct communication, information about acceleration can be transmitted directly. Second, information that a car obtains from its own sensors (e.g. wheel encoders for velocity, accelerometer for acceleration) is more accurate than information a car derives from its radar input.

The task of our safety controller is to determine what the minimal safe time headway is, with the constraint that it still must be safe. We define the minimal safe time headway as follows: it is the time headway a car must keep from its preceding vehicle at which the probability of a crash does not exceed 0.1%. This definition can easily be changed to a lower or higher number, using the same model. We get back to this in Section IV-B.

All controllers share the same update scheme, that is based on a simple physics model in which velocity is updated according to acceleration, after which position is updated according the velocity. This update scheme is depicted in Algorithm 1. Note that in our models we denote velocity as the first derivative of position, $\dot{x}$, and the acceleration as the second derivative of position, $\ddot{x}$.

All controllers use this simple update scheme. The difference between the ACC and the two CACC controllers lies in the way the controller derives the acceleration of the preceding vehicle. The ACC controller does this using radar, while the CACC controllers directly transmits information about their own acceleration to the following vehicle. In the following paragraphs, the controllers are described in detail. These descriptions are at the agent level.

```
/* Δt = 0.01                                        */
foreach timestep t do
    ẍ_t ← compute new ẍ;
    ẋ_t ← ẋ_{t−1} + ẍ_t Δt;
    x_t ← x_{t−1} + ẋ_t;
end
```

**Algorithm 1**: Update scheme for cars

*a) ACC controller:* The radar in the ACC controller has an update rate of 10Hz, and it takes an additional 5ms to process each measurement. Each measurement consists of two values: the range (i.e. the distance to the preceding vehicle, measured in meters) and range rate (i.e. the relative velocity to the preceding vehicle). Using two consecutive range rate measurements, the acceleration of the preceding vehicle can be derived.

In Algorithm 2, the pseudocode for the behaviour of this controller is given.

```
foreach timestep t do
    if Processed radar measurement m then
        ẋ_{preceding,t} ← m.rangeRate + ẋ_{self};
        ẍ_{preceding,t} ← ẋ_{preceding,t} − ẋ_{preceding,t−1};
        changeAcceleration(ẍ_{preceding,t});
    end
end
```

**Algorithm 2**: Behaviour of the ACC controller

This algorithm takes the range rate from each radar measurement, computes the acceleration of the preceding vehicle and changes its own acceleration according to this acceleration. No communication between vehicles occurs in this setting.

*b) CACC1 controller:* Both CACC controllers use direct vehicle-to-vehicle (V2V) communication to exchange messages contains the vehicle's acceleration. These messages are sent asynchronously by each car at a frequency of 10Hz.

The CACC1 controller sends the measured value from the accelerometer to the following vehicle. This means that the value in this message can directly be used by the following vehicle, instead of having to derive the acceleration from multiple radar readings. This makes the CACC1 controller more responsive than the ACC controller. In Algorithm 3, the pseudocode for the behaviour of this controller is given.

This algorithm copies the acceleration value that it received from the preceding vehicle. Also, this controller makes sure that that each car sends messages containing its own acceleration (measured from the accelerometer) to the following vehicle.

*c) CACC2 controller:* The CACC2 controller is a simple extension of the CACC1 controller. Instead of the value of the accelerometer, the *predicted* value of the acceleration is communicated to the following vehicle. Since there is a delay between a braking action and the actual deceleration of the

```
foreach timestep t do
    if Received message m then    // from preceding car
        ẍ_{preceding,t} ← m.ẍ;
        changeAcceleration(ẍ_{preceding,t});
    end
    if Sender is ready then                      // 10Hz
        sendMessage(ẍ_{actual});         // to following car
    end
end
```

**Algorithm 3**: Behaviour of the CACC1 controller

vehicle of 150ms, and the vehicle is able to predict the actual deceleration very well at the time of the braking action, this is a very sensible thing to do.

In Algorithm 4, the pseudocode for the behaviour of this controller is given.

```
foreach timestep t do
    if Received message m then    // from preceding car
        ẍ_{preceding,t} ← m.ẍ;
        changeAcceleration(ẍ_{preceding,t});
    end
    if Sender is ready then                      // 10Hz
        sendMessage(ẍ_{predicted});      // to following car
    end
end
```

**Algorithm 4**: Behaviour of the CACC2 controller

This algorithm is very similar to the CACC1 algorithm. The only subtle difference is that, instead of the measured acceleration from the accelerometer, the predicted acceleration is communicated the following vehicle. This makes the CACC2 controller more responsive than the CACC1 controller.

Both CACC controllers use time-based communication as opposed to event-based communication. On the one hand, this makes our model more robust, since receiving vehicles know when to expect new messages and are able to anticipate when a message does not arrive. This is not the case when we use event-based communication. On the other hand, time-based communication is less scalable. When many cars within a certain range are constantly broadcasting messages, this will influence the quality of the network.

### *B. Uncertainty & Delay*

The algorithms that we described in the previous sections are fairly straightforward, and if one knows the delays of communication and information processing, it would seem easy to compute the minimal safe time headway cars should maintain to avoid crashes. However, since sensor information and wireless communication comes with a lot of uncertainty, the computation of the minimal safe time headway is not a trivial task. In this section, we describe the various factors of uncertainty that are present in our sensors, and the values we used for these uncertain parameters. These values can be seen as realistic. However, if the values of these parameters change, we can still use the same model to approximate the safe time headway that belongs to that parameter set. We will discuss this issue further in Section V.

We have modelled all uncertainties using a Gaussian distribution with deviation $\sigma$ from the correct value $\mu$. The delays are hard-coded in the model.

Uncertainty:

- Radar range rate: $\sigma = 0.1$m/s. This is the relative velocity measurement of the radar. This influences the computation of the preceding car's deceleration in the ACC vehicles;
- Failure in radar range rate: in $0.1\%$ of the radar measurements, the radar fails to measure the relative velocity of the preceding vehicle;
- Own velocity: $\sigma = 0.1$m/s. This also influences the computation of the preceding car's deceleration in the ACC vehicles;
- Own max braking power: $\sigma = 0.3$m/s$^2$ one-sided. This is the error in a car's estimation of its own maximum braking power. For example, it could be that a car thinks it can brake with $-9$m/s$^2$, while in reality this is only $8.7$m/s$^2$. It is one-sided, since $-9$m/s$^2$ is a car's maximum braking power. This influences the CACC2 messages with the predicted braking power;
- Own predicted acceleration: $\sigma = 0.3$m/s$^2$. This is the uncertainty of the estimation of the acceleration when a braking action occurs. This is the value that is sent by the CACC2 vehicles, before the deceleration actually occurs;
- Own accelerometer value: $\sigma = 0.2$m/s$^2$. In the CACC1 controller, the car only sends out the estimation of its own acceleration. This has slightly less uncertainty than the predicted acceleration;
- Failure in broadcasting: About $1\%$ of all sent messages do not arrive at their destination. This is a simplification, because currently, we do not take into account bursts of packet loss.
- Radar range measurement: $\sigma = 0.5$m. The distance to the preceding vehicle. This measurement is not used in our simulation, but we do include it in our calculations to determine the minimal safe distance. See Section V for more detail.

Delay:

- Mechanical brake delay: 150ms. This is the delay between the change in acceleration and the actual start of that acceleration. This is essentially the difference between CACC1 and CACC2.
- Radar processing delay: 50ms. This is how long it takes before a radar signal is processed and ready to use.
- Communication delay: 10ms. This is how long it takes for a message to be received.

These are the uncertainties and delays that we incorporated in our experiments. Others include delay in bus, gateway and radio channel access. These are not included in our experiments, but these could easily be incorporated.

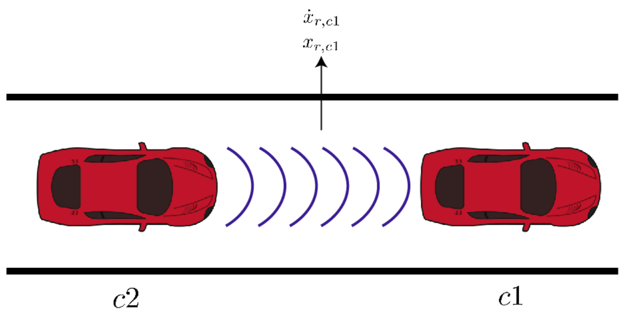


Fig. 1. The ACC scenario, in which car $c2$ uses its radar to obtain information about relative position $x_{r,c1}$ and velocity $\dot{x}_{r,c1}$ of car $c1$.

## IV. Experiments & Results

In this section, we describe the scenario that we implemented and the experiments we performed in this scenario.

### A. Scenario

The objective of our experiments is to find a minimal safe spacing policy for different adaptive cruise control controllers, given various parameters that describe factors of uncertainty in the system, as well as the vehicle's velocity.

To this end, we simulate 2 cars on a highway, of which the front vehicle applies an emergency brake at $t_0$. The following car has to respond to this emergency brake. The way the following car responds to the emergency brake, depends on the controller in the car. The ACC controller uses radar, the CACC1 controller uses direct communication of the actual deceleration, and the CACC2 controller uses direct communication of the predicted deceleration. In Figures 1 and 2, these scenarios are illustrated.

In our experiments, we use a homogeneous set of cars. This is a simplification of real situations on highways, where there are of course many different kinds of vehicles with different capabilities. This model can however be used with *any* vehicle model, and *any* model of uncertainty. Our experiments are an example of how this model can be used to determine minimal safe time headways.

We did simulations with two different initial velocities, 20m/s and 30m/s. We then measured the difference in displacement:

$$\Delta s_{j,i} = s_j - s_i \tag{1}$$

where $s_i$ is the displacement of the front vehicle and $s_j$ the displacement of the following vehicle.

The displacement of a vehicle $i$ is the distance that $i$ travelled from $t_0$ to the moment of standstill $t_{ss}$:

$$s_i = x_{i,t_{ss}} - x_{i,t_0} \tag{2}$$

We did experiments with 3 controllers (ACC, CACC1 and CACC2) $\times$ 2 initial velocities (20m/s and 30m/s) $\times$ 100.000 runs per setting $=$ 600.000 runs in total.

For all these runs, we measure the difference in displacement of the cars, $\Delta s$. For each combination of a controller

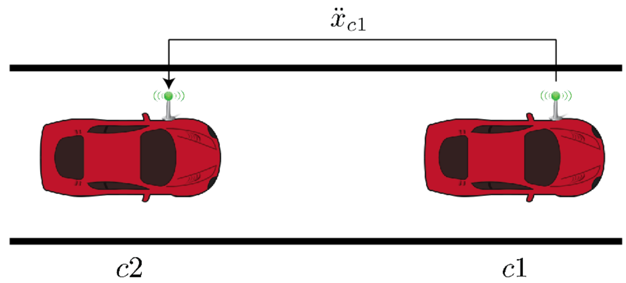


Fig. 2. The CACC scenario, in which car $c1$ uses direct communication to transmit its acceleration $\ddot{x}_{c1}$ to the following vehicle.

TABLE I
THE VALUES OF $\Delta s_{\text{SAFE}}$ FOR EACH BASE DISTRIBUTION $\mathcal{B}$.

| Distribution | $\Delta s_{\text{safe}}$ |
|---|---|
| $\mathcal{B}_{\text{ACC},20}$ | 10.53m |
| $\mathcal{B}_{\text{CACC1},20}$ | 7.91m |
| $\mathcal{B}_{\text{CACC2},20}$ | 4.81m |
| $\mathcal{B}_{\text{ACC},30}$ | 17.42m |
| $\mathcal{B}_{\text{CACC1},30}$ | 12.17m |
| $\mathcal{B}_{\text{CACC2},30}$ | 7.69m |

with an initial velocity, we then obtain the distributions of $\Delta s$. For the remainder of this paper, we will call these the *base distributions* $\mathcal{B}_{c,\dot{x}}$, with $c \in \{\text{ACC, CACC1, CACC2}\}$ and $\dot{x}$ being the initial velocity of the vehicles, and in our experiments, $\dot{x} \in \{20, 30\}$.

### B. Safe spacing policy approximation

In this section, we explain how we can use the base distributions $\mathcal{B}_{c,\dot{x}}$ to approximate the minimal safe time headway. In order to make these calculations, some variables must be introduced:

- $h_{\dot{x},\text{safe}}$: the minimal safe time headway, given velocity $\dot{x}$. This is what we want to compute;
- $p(Crash)$: the acceptable probability for a crash. We use a value of 0.1%;
- $\Delta s_{\dot{x},\text{safe}}$: the safe difference in displacement, given velocity $\dot{x}$. This value depends on $\dot{x}$ and on the value of $p(Crash)$;
- $d_{i,j}$: the final distance between vehicle $i$ and $j$ at standstill, after a run;

For each distribution $\mathcal{B}_{c,\dot{x}}$, we can compute the value of $\Delta s_{\text{safe}}$. In order to do this, we first compute the convolution of the base distributions with the uncertainty on the radar range. This is necessary, because the safe distance is dependent on how well a car is able to determine its distance to the preceding vehicle. Then, we determine $\Delta s_{\text{safe}}$ by looking up at which value of $\Delta s$, the right-hand side tail of the distribution consists of $p(Crash)\%$ of the runs. In our experiments, we set the desired probability of a crash at 0.1%, but this can easily be set to a different value without having to redo the experiments. In this case, only this final step needs to be recalculated.

Using these values, we can now compute the minimal safe time headway $h_{\dot{x},\text{safe}}$, by dividing these values by the initial velocity of the vehicles.

TABLE II
THE VALUES OF $h_{\dot{x},\text{SAFE}}$ FOR EACH BASE DISTRIBUTION $\mathcal{B}$.

| Distribution | $h_{\dot{x},\text{safe}}$ |
|---|---|
| $\mathcal{B}_{\text{ACC},20}$ | 0.53s |
| $\mathcal{B}_{\text{CACC1},20}$ | 0.40s |
| $\mathcal{B}_{\text{CACC2},20}$ | 0.24s |
| $\mathcal{B}_{\text{ACC},30}$ | 0.58s |
| $\mathcal{B}_{\text{CACC1},30}$ | 0.41s |
| $\mathcal{B}_{\text{CACC2},30}$ | 0.26s |

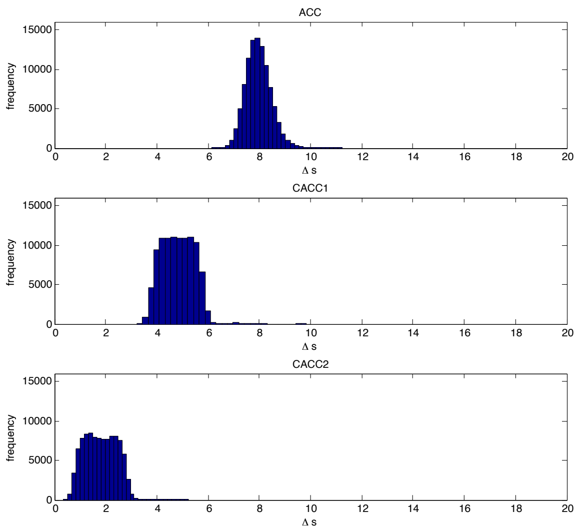


Fig. 3. Results for our controllers when using initial velocity $v = 20$m/s.

$$h_{\dot{x},\text{safe}} = \frac{\Delta s_{\dot{x},\text{safe}}}{\dot{x}} \quad (3)$$

### C. Results

The results of our simulations are visualised in Figures 3 and 4, that show respectively the experiments with initial velocity $\dot{x} = 20$m/s and $\dot{x} = 30$m/s. The further to the right on the horizontal axis the distributions are, the higher the distance is that the following vehicle has travelled, relative to the leading vehicle. So, the further the distributions are to the left on the horizontal axis, the lower the minimal safe time headway will be for the following vehicles.

Using the safe spacing policy approximation method as described in the previous section, we obtained for each base distribution $\mathcal{B}$ the safe distance $\Delta s_{\dot{x},\text{safe}}$ and the safe time headway $h_{\dot{x},\text{safe}}$ given a certain initial velocity $\dot{x}$. These results are summerised in Tables I and II.

## V. ANALYSIS

In this section, we analyse the results that we presented in the previous section, and we validate these results by doing control experiments with the results that we found.

### A. Table analysis

The main results of our work are summarised in Tables I and II. We see that the differences in safe time headway are very small for different initial velocities. This means that safe time headway is (more or less) independent of initial velocity.

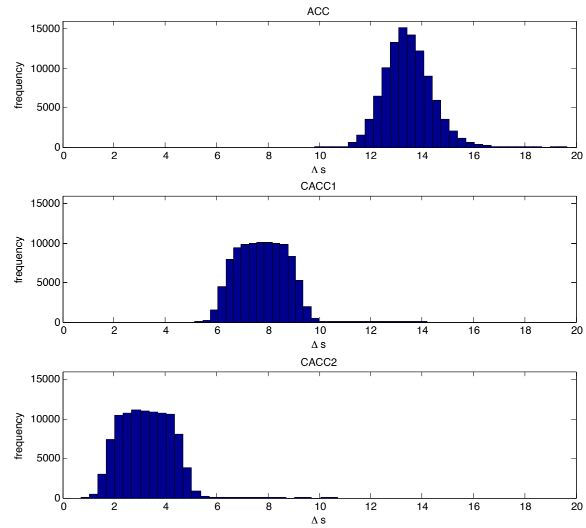


Fig. 4. Results for our controllers when using initial velocity $v = 30$m/s.

The results for safe time headway are very promising in terms of feasibility. The numbers that we obtained are the absolute minimum time headway that cars should keep. However, research into string stability has shown that ideal time headways are usually larger than the numbers that we found. For example, research on string stability by Naus *et al.* [14] report on minimal time headways of 2.8s for ACC systems and 0.8s for CACC systems. These numbers are much bigger than our results for the safety controller. This is good news, because in this case, the string stability controller will not often be interfered by our safety controller.

### B. Base distribution analysis

In Figures 3 and 4, we show the histograms that contain the distributions of our measured variable, $\Delta s$. These are the *base distributions* $\mathcal{B}_{c,\dot{x}}$, that we introduced in Section IV. The higher the $\Delta s$, the higher the relative displacement of the following vehicle. This means that lower values for $\Delta s$ are better.

On an abstract level, we can draw the following conclusions about the controllers:

- As expected, the ACC controller performs the worst, and the CACC2 controller performs the best.
- There is a difference in the shape of the distributions between the ACC controller and the two CACC controllers: The CACC distributions are flatter than the ACC distribution. This is due to the uncertainty and delay that are introduced with wireless communication.
- When comparing Figure 3 to Figure 4, we see a relation between the distributions of the same controller. The means and standard deviations are both larger for a larger initial velocity. However, we tried to fiend a direct relation between these distributions, but we could not find such a relation. Further research is needed in order to achieve

TABLE III
CRASH STATISTICS IN OUR CONTROL EXPERIMENTS.

| Distribution | % crash | mean crash velocity | max crash velocity |
|---|---|---|---|
| $\mathcal{B}_{\text{ACC},20}$ | 0.09% | 1.66m/s | 3.02m/s |
| $\mathcal{B}_{\text{CACC1},20}$ | 0.07% | 2.07m/s | 4.60m/s |
| $\mathcal{B}_{\text{CACC2},20}$ | 0.1% | 1.55m/s | 3.55m/s |
| $\mathcal{B}_{\text{ACC},30}$ | 0.09% | 2.53m/s | 5.50m/s |
| $\mathcal{B}_{\text{CACC1},30}$ | 0.07% | 1.42m/s | 2.95m/s |
| $\mathcal{B}_{\text{CACC2},30}$ | 0.1% | 1.80m/s | 3.39m/s |

this. This means that as of now, we cannot create a base distribution in which we leave out the initial velocity as well.

### C. Validation

In this section, we will validate the results in Table II. We will do this by running our simulations once again, but now using the initial headway values from that table. In Figure 5, the histograms of the final distances between the two vehicles are shown. Negative values on the horizontal axis denote that a crash has occurred. From these histograms, it becomes immediately apparent that only a very small portion of the runs result in a crash.

Table III shows statistics about the crashes that occurred in these simulations. We measured the percentage of crashes that occurred, and we measured the mean velocity of the crashes, as well as the velocity of the fastest crash in each setting. From these results, we see that for all controllers, the probability for a crash is indeed $0.1\%$, or even a little bit less. This validates the values for safe time headway that we obtained from the base distributions $\mathcal{B}$.

Apart from the probability for a crash, we also recorded the mean and maximum velocity of the crashes. We can conclude from this table that *none* of the crashes that occurred in the control experiments are lethal. The worst crash that we found was 5.5m/s. This means that at the time of the crash, the following car was driving 5.5m/s faster than the preceding vehicle. This type of crash is easily damped by seatbelts and the airbag.

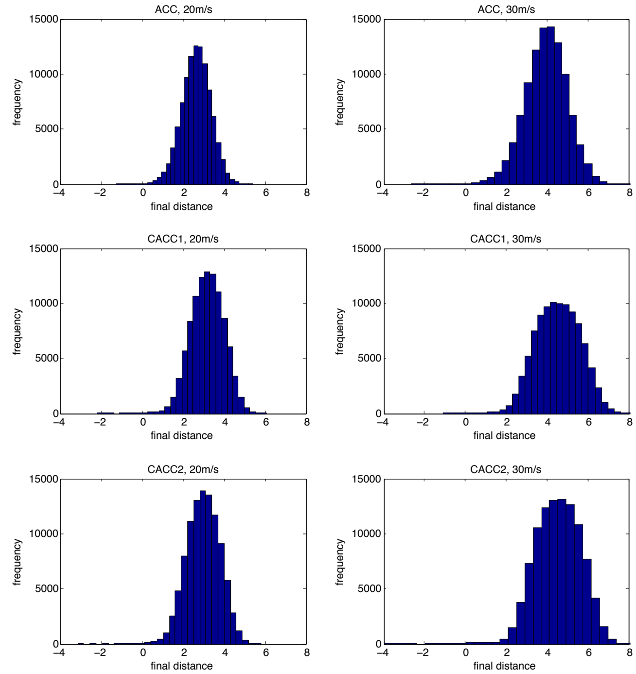


Fig. 5. Results of our validation experiments. Using the safe time headway values from Table II, these are the distributions of final distances from our simulations.

## VI. CONCLUSION

In this paper, we looked at safe minimal time headways for (Cooperative) Adaptive Cruise Control (CACC) systems. Whereas much research effort is spent on making this system such that they are comfortable for the driver (e.g. that the vehicle does not continuously and abruptly accelerates and brakes), surprisingly less effort goes into making these systems safe.

One of the major development transitions in road of commercialising ACC systems, was that it was shown that variable time headways had to be determined dynamically instead of beforehand. In this light, an important contribution of this paper is that it demonstrates that the same holds for CACC systems. Moreover, because of the many uncertainties involved with communicating information between vehicles, the urge for dynamic variable time headways is even stronger.

Our investigation has been experimental by means of computer simulation. These experiments show that the resulting safe time headways are more or less independent of initial velocity. Also, the found values for minimal safe time headway are lower than the values that are currently used in controllers that optimise on string stability. This means that our safety controller will not interfere with the string stability controller under normal circumstances.

Our resulting safe time headways still have a small, calculated risk of crashing. But even in the very unlikely event of a crash ($< 0.1\%$), the crashes that occur are far from lethal.

The work described in this paper is a first step towards enabling vehicles to fully determine (variable) spacing policies on-line (i.e. while driving) and autonomously. For future work, the next step is to translate the findings of our experimental investigation into letting vehicles determine time headways while driving. On the experimental side, our simulations were designed with minimal requirements for the strict purpose of isolating the investigation of the research question. Still, we also propose scaling up the simulations in the future in terms of numbers of vehicles and complexity of the road network.

## REFERENCES


[1] D. Swaroop and K. Rajagopal, “A review of constant time headway policy for automatic vehicle following,” in *Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE*, 2001, pp. 65 –69.

[2] P.Petrov, “Nonlinear adaptive control of a two-vehicle convoy,” *The Open Cybernetics and Systemics Journal*, vol. 3, pp. 70–78, 2009.

[3] D. Swaroop and J. Hedrick, “Direct adaptive longitudinal control of vehicle platoons,” in *Decision and Control, 1994., Proceedings of the 33rd IEEE Conference on*, vol. 1, dec 1994, pp. 684 –689 vol.1.

[4] K. Santhanakrishnan and R. Rajamani, "On spacing policies for highway vehicle automation," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 4, no. 4, pp. 198 – 204, dec. 2003.

[5] J. Wang and R. Rajamani, "The impact of adaptive cruise control systems on highway safety and traffic flow," *Proceedings of The Institution of Mechanical Engineers Part D-journal of Automobile Engineering*, vol. 218, pp. 111–130, 2004.

[6] ——, "Should adaptive cruise-control systems be designed to maintain a constant time gap between vehicles?" *Vehicular Technology, IEEE Transactions on*, vol. 53, no. 5, pp. 1480 – 1490, sept. 2004.

[7] J. Zhao, M. Oya, and A. El Kamel, "A safety spacing policy and its impact on highway traffic flow," in *Intelligent Vehicles Symposium, 2009 IEEE*, june 2009, pp. 960 –965.

[8] J. Yi and R. Horowitz, "Macroscopic traffic flow propagation stability for adaptive cruise controlled vehicles," *Transportation Research*, vol. 14, no. 2, pp. 81–95, April 2006.

[9] J. Zhang and P. Ioannou, "Adaptive vehicle following control system with variable time headways," in *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, dec. 2005, pp. 3880 – 3885.

[10] B. van Arem, C. van Driel, and R. Visser, "The impact of cooperative adaptive cruise control on traffic-flow characteristics," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 7, no. 4, pp. 429 –436, 2006.

[11] X. Yang, J. Liu, F. Zhao, and N. H. Vaidya, "A vehicle-to-vehicle communication protocol for cooperative collision warning," *Mobile and Ubiquitous Systems, Annual International Conference on*, vol. 0, pp. 114–123, 2004.

[12] A. Broggi, M. Bertozzi, A. Fascioli, C. G. L. Bianco, and A. Piazzi, "Visual perception of obstacles and vehicles for platooning," *IEEE TRANS. INTELL. TRANSPORT. SYS*, vol. 1, no. 3, pp. 164–176, 2000.

[13] I. Kanellakopoulos, P. Nelson, and O. Stafsudd, "Intelligent sensors and control for commercial vehicle automation," *Annual Reviews in Control*, vol. 23, pp. 117 – 124, 1999. [Online]. Available: http://www.sciencedirect.com/science/article/B6V0H-482MDS0-15/2/ed4a9b9020ab72601347ead8b50fd0c8

[14] G. Naus, R. Vugts, J. Ploeg, M. van de Molengraft, and M. Steinbuch, "String-stable cacc design and experimental validation: A frequency-domain approach," *Vehicular Technology, IEEE Transactions on*, vol. 59, no. 9, pp. 4268 –4279, 2010.

[15] S. Hallé, "Automated highway systems: Platoons of vehicles viewed as a multiagent system," Master's thesis, Université Laval, Québec, 2005.

[16] M. Khan, D. Turgut, and L. Bölöni, "A study of collaborative influence mechanisms for highway convoy driving," in *Proceedings of International Workshop on Agents in Traffic and Transportation (ATT08), in conjunction with the Seventh Joint Conference on Autonomous and Multi-Agent Systems (AAMAS 2008)*, May 2008, pp. 46–53.

[17] W. van Willigen, M. Schut, and L. Kester, "Evaluating adaptive cruise control strategies in worst-case scenarios," in *To appear in: Proceedings of the 14th International IEEE Conference on Intelligent Transportation Systems - (ITSC 2011)*, October 2011.